

A 1.75 LP approximation for the Tree Augmentation Problem

Guy Kortsarz¹ and Zeev Nutov²

¹ Rutgers University–Camden, NJ. guyk@camden.rutgers.edu

² The Open University of Israel. nutov@openu.ac.il.

Abstract. In the Tree Augmentation Problem (TAP) the goal is to augment a tree T by a minimum size edge set F from a given edge set E such that $T \cup F$ is 2-edge-connected. The best approximation ratio known for TAP is 1.5. In the more general **Weighted TAP** problem, F should be of minimum weight. **Weighted TAP** admits several 2-approximation algorithms w.r.t. to the standard cut LP-relaxation, but for all of them the performance ratio of 2 is tight even for TAP. The problem is equivalent to the problem of covering a laminar set family. Laminar set families play an important role in the design of approximation algorithms for connectivity network design problems. In fact, **Weighted TAP** is the simplest connectivity network design problem for which a ratio better than 2 is not known. Improving this “natural” ratio is a major open problem, which may have implications on many other network design problems. It seems that achieving this goal requires finding an LP-relaxation with integrality gap better than 2, which is a long time open problem even for TAP. In this paper we introduce such an LP-relaxation and give an algorithm that computes a feasible solution for TAP of size at most 1.75 times the optimal LP value. This gives some hope to break the ratio 2 for the weighted case. Our algorithm computes some initial edge set by solving a partial system of constraints that form the integral edge-cover polytope, and then applies local search on 3-leaf subtrees to exchange some of the edges and to add additional edges. Thus we do not need to solve the LP, and the algorithm runs roughly in time required to find a minimum weight edge-cover in a general graph.

1 Introduction

1.1 Problem definition and related problems

A graph (possibly with parallel edges) is *k-edge-connected* if there are k pairwise edge-disjoint paths between every pair of its nodes. We study the following fundamental connectivity augmentation problem: given a connected undirected graph $G = (V, \mathcal{E})$ and a set of additional edges (called “links”) E on V disjoint to \mathcal{E} , find a minimum size edge set $F \subseteq E$ such that $G + F = (V, \mathcal{E} \cup F)$ is 2-edge-connected. The 2-edge-connected components of the given graph G form a tree. It follows that by contracting these components, one may assume that G is a tree. Hence, our problem is:

Tree Augmentation Problem (TAP)

Instance: A tree $T = (V, \mathcal{E})$ and a set of links E on V disjoint to \mathcal{E} .

Objective: Find a minimum size subset $F \subseteq E$ of links such that $T \cup F$ is 2-edge-connected.

TAP can be formulated as the problem of covering a laminar set family as follows. Root T at some node r . Every edge of T partitions T into two parts T' and $T \setminus T'$, where $r \notin T'$; let $\hat{\mathcal{E}}$ denote the set family obtained by picking for each edge the part T' that does not contain r . Then $\hat{\mathcal{E}}$ is laminar, and $F \subseteq E$ is a feasible solution for TAP if and only if F covers $\hat{\mathcal{E}}$, namely, for every $T' \in \hat{\mathcal{E}}$ there is a link in F from T' to $T \setminus T'$. TAP is also equivalent to the problem of augmenting the edge-connectivity from k to $k+1$ for any odd k ; this is since the family of minimum cuts of a k -connected graph with k odd is laminar.

In the more general **Weighted TAP** problem, the links in E have weights $\{w_e : e \in E\}$ and the goal is to find a minimum weight augmenting edge set $F \subseteq E$ such that $T \cup F$ is 2-edge connected. Even a more general problem is the **2-Edge-Connected Subgraph** problem, where the goal is to find a spanning 2-edge-connected subgraph of a given weighted graph; **Weighted TAP** is a particular case, when the input graph contains a connected spanning subgraph of cost zero.

1.2 Previous and related work

TAP is NP-hard even for trees of diameter 4 [9], or when the set E of links forms a cycle on the leaves of T [2]. The first 2-approximation for **Weighted TAP** was given 24 years ago in 1981 by Fredrickson and Jájá [9], and was simplified later by Khuller and Thurimella [15]. These algorithms reduce the problem to the **Min-Cost Arborescence** problem, that is solvable in polynomial time [4,6]. The primal-dual algorithm of [12,11] is another combinatorial 2-approximation algorithm for the problem. The iterative rounding algorithm of Jain [13] is an LP-based 2-approximation algorithms. The approximation ratio of 2 for all these algorithms is tight even for TAP. These algorithms achieve ratio 2 w.r.t. to the standard *cut LP* that seeks to minimize $\sum_{e \in E} w_e x_e$ over the following polyhedron:

$$\left\{ x \in \mathbb{R}^E : x(\delta(A)) \geq 1 \ \forall A \in \hat{\mathcal{E}}, x_e \geq 0 \ \forall e \in E \right\} .$$

Here $\delta(A)$ is the set of links with exactly one endnode in A and $x(F) = \sum_{e \in F} x_e$ is the sum of the variables indexed by the links in F .

Laminar set families play an important role in the design and analysis of exact and approximation algorithms for network design problems, both in the primal-dual method and the iterative rounding method, c.f. [18,12]. **Weighted TAP** is the simplest network design problem for which a ratio better than 2 is not known. Breaking the “natural” ratio of 2 for **Weighted TAP** is a major open problem in network design, which may have implications on many other problems, c.f. the surveys [14,16].

As a starting point, Khuller [14] in his survey on high connectivity network design problems posed as a major open question achieving ratio better than 2

for TAP. This open question was resolved by Nagamochi [20], that used a novel lower bound to achieve ratio $1.875 + \epsilon$ for TAP. Building on the lower bound idea of Nagamochi [20], the sequence of papers [7,8,17] introduced additional new techniques to achieve ratio 1.8 by a much simpler algorithm and analysis, and also achieved the currently best known ratio 1.5 by a more complex algorithm.

Several algorithms for **Weighted TAP** with ratio better than 2 are known for special cases. Cheriyan, Jordán, and Ravi [2] showed how to round a half-integral solution to the cut LP within ratio $4/3$. However, as is pointed in [2], there are TAP instances that do not have an LP optimal solutions which is half integral. In [5] is given an algorithm with ratio $(1 + \ln 2)$ and running time $n^{f(D)}$ where D is the diameter of T .

Studying various LP-relaxations for TAP is motivated by the hope that these may lead to breaking the ratio of 2 for **Weighted TAP**. Thus several paper analyzed integrality gaps of LP relaxations for the problem. Cheriyan, Karloff, Khanderkar, and Koenemann [3] showed that the integrality gap of the standard cut LP is at least 1.5 even for TAP. For the special case of TAP when every link connects two leaves, [19] obtained ratios $5/3$ w.r.t. the cut LP, ratio $3/2$ w.r.t. to a strengthened “leaf edge-cover” LP, and ratio $17/12$ not related to any LP. However, the analysis of [19] does not extend to the general TAP. In this paper, with the help of some ideas from [19,8,17,1], we introduce a new LP-relaxation and prove that its integrality gap at most $7/4$ for TAP. This gives some hope to break the ratio 2 for the weighted case.

Finally, we mention some work on the closely related **2-Edge-Connected Subgraph** problem. This problem was also vastly studied. For general weights, the best known ratio is 2 by Fredrickson and Jájá [9], which can also be achieved by the algorithms in [15] and [13]. For particular cases, better ratios are known. Fredrickson and Jájá [10] showed that when the edge weights satisfy the triangle inequality, the Christofides heuristic has ratio $3/2$. For the special case when all the edges of the input graph have unit weights (the “min-size” version of the problem), the currently best known ratio is $4/3$ [22].

2 A new LP-relaxation for TAP

In this section we introduce a new LP-relaxation for TAP, and in subsequent section prove a $7/4$ integrality gap (for the unweighted case). This LP-relaxation combines ideas from [19,8,17,1], but also introducing new crucial valid constraints. Later, we will also introduce a technique of exchanging fractional values between the edges. We are not aware of such methods used to prove an integrality gap. We need some definition to introduce our LP.

For $u, v \in V$ let $(u, v) \in \mathcal{E}$ denote the edge in T and uv the link in E between u and v . A link uv *covers* all the edges along the path $P(uv)$. The choice of the root r defines a partial order on V : u is a *descendant* of v (or v is an *ancestor* of u) if v belongs to $P(ru)$; if, in addition, $(u, v) \in T$, then u is a *child* of v , and v is the *parent* of u . The *leaves* of T are the nodes in $V \setminus \{r\}$ that have no descendants. We denote the leaf set of T by $L(T)$, or simply by L , when the

context is clear. The *rooted subtree* of T induced by r' and its descendants is denoted by $T_{r'}$ (r' is the root of $T_{r'}$). A subtree T' of T is called a *rooted subtree* of T if $T' = T_{r'}$ for some $r' \in V$.

Definition 1 (shadow, shadow-minimal cover). Let $P(uv)$ denote the path between u and v in T . A link $u'v'$ is a *shadow* of a link uv if $P(u'v') \subseteq P(uv)$. A cover F of T is *shadow-minimal* if for every link $uv \in F$ replacing uv by any proper shadow of uv results in a set of links that does not cover T .

Every TAP instance can be rendered closed under shadows by adding all shadows of existing links. We refer to the addition of all shadows as *shadow completion*. Shadow completion does not affect the optimal solution size, since every shadow can be replaced by some link covering all edges covered by the shadow. Thus we may assume that the set of links E is closed under shadows.

Definition 2 (twin-link, stem). A link between two leaves a, b of T is a *twin-link* if its contraction results in a new leaf; a, b are called *twins* and their least common ancestor is called a *stem*. Let W denote the set of twin links, and for $e \in W$ let s_e denote the stem of a twin-link e .

For $A, B \subseteq V$ and $F \subseteq E$ let $\delta_F(A, B)$ denote the set of links in F with one end in A and the other end in B , and let $\delta_F(A) = \delta_F(A, V \setminus A)$ denote the set of links in F with exactly one endnode in A . The default subscript in the above notation is E . Let L denote the set of leaves of T and let

$$\mathcal{O}_L = \{A \subseteq V : |A \cap L| \text{ is odd}\}.$$

For a function x on E and $F \subseteq E$ let $x(F) = \sum_{e \in F} x_e$.

Let Π be the polyhedron defined by the following set of linear constraints:

$$x(\delta(A)) \geq 1 \quad \forall A \in \hat{\mathcal{E}} \quad (1)$$

$$x(\delta(A, V)) \geq \lceil |A \cap L|/2 \rceil \quad \forall A \in \mathcal{O}_L \quad (2)$$

$$x(\delta(v)) = 1 \quad \forall v \in L \quad (3)$$

$$x_e - x(\delta(s_e)) = 0 \quad \forall e \in W \quad (4)$$

$$x_e \geq 0 \quad \forall e \in E \quad (5)$$

Inequalities (1) and (5) are the constraints of a standard LP-relaxation for TAP. Inequality (2) were used in [19] to establish integrality gap of 1.5 for the special case of TAP when every link connects two leaves. We add over it the constraints (3) and (4), which are crucial to obtain integrality gap better than 2 for TAP.

Now we explain why the above LP is a relaxation for TAP.

Definition 3 (exact cover). An edge set F is an *exact cover* of L if $|\delta_F(v)| = 1$ for all $v \in L$.

Lemma 1 ([17]). Given a TAP instance with shadow completion, let F be an optimal shadow-minimal solution with $|F \cap W|$ maximal. Then F is an exact cover of L , and for any $e = ab \in W$, either $e \in F$ and $|\delta_F(s_e)| = 1$, or $e \notin F$ and $|\delta_F(s_e)| = 0$. \square

Let Π_L be the polyhedron defined by (2), (3), and (5). Then Π_L is the convex hull of the exact edge-covers of L , see [21, Theorem 34.2]; thus by Lemma 1, these constraints are valid. The validity of the constraints (4) follows also from Lemma 1. Consequently, the linear program $\tau = \min\{x(E) : x \in \Pi\}$ is a relaxation for TAP. Combining techniques from [8,17] and using some new methods, we prove the following.

Theorem 1. *TAP admits a polynomial time algorithm that computes a solution F such that $|F| \leq \frac{7}{4}\tau$.*

3 Proof of Theorem 1

3.1 Reduction to the minimum weight leaf edge-cover problem

Let $S = \{s_e : e \in W\}$ be the set of stems of T and let $R = V \setminus (L \cup S)$. Let $\rho \geq 1.5$ be a parameter, set later to $\rho = 7/4$. Define a weight function w on $E(L, V)$ by:

$$w_e = \begin{cases} \rho & \text{if } e \in \delta(L, L) \setminus W \\ \rho - \frac{1}{2} & \text{if } e \in \delta(L, V \setminus L) \\ \rho + \frac{1}{2} & \text{if } e \in W \end{cases}$$

Lemma 2. *Let F_L be a minimum weight exact cover of L and $x \in \Pi$ such that $x(E) = \tau$. Then:*

$$\rho\tau \geq w(F_L) + \frac{1}{2} \sum_{v \in R} x(\delta(v)). \quad (6)$$

Proof. Let x' be defined by $x'_e = x_e$ if $e \in \delta(L, V)$ and $x'_e = 0$ otherwise. Note that $x' \in \Pi_L$, since x satisfies (2), (3), and (5). Since F_L is an optimal (integral) exact cover of L with respect to the weights w_e and $x' \in \Pi_L$, we have:

$$x' \cdot w \geq w(F_L) .$$

Thus to prove the lemma, it is sufficient to prove the following:

$$\rho\tau \geq x' \cdot w + \frac{1}{2} \sum_{v \in R} x(\delta(v)) .$$

Assign ρx_e tokens to every $e \in E$. The total amount of tokens is exactly $\rho x(E) = \rho\tau$. We will show that these tokens can be moved around such that the following holds:

- (i) Every $e \in \delta(L, L)$, and thus every $e \in W$, keeps its initial ρx_e tokens.
- (ii) Every $e \in \delta(L, V \setminus L)$ keeps $(\rho - \frac{1}{2})x_e$ tokens from its initial ρx_e tokens.
- (iii) Every $e \in W$ gets additional $\frac{1}{2}x_e$ tokens, to a total of $(\rho + \frac{1}{2})x_e$ tokens.
- (iv) Every $v \in R$ gets $\frac{1}{2}x_e$ tokens for each $e \in \delta(v)$.

This distribution of tokens is achieved in two steps. In the first step, for every $e \in E$, move $\frac{1}{2}x_e$ token from the ρx_e tokens of e to each non-leaf endnode of e , if any. Note that after this step, (i) and (ii) hold, and every $v \in V \setminus L$ gets $\frac{1}{2}x_e$ token for each $e \in \delta(v)$. In the second step, every $e \in W$ gets all the tokens moved at the first step to its stem s_e by the links in $\delta(s_e)$. The amount of such tokens is $\frac{1}{2}x(\delta(s_e)) = \frac{1}{2}x_e$, by (4). This gives an assignment of tokens as claimed, and thus the proof of the lemma is complete. \square

To prove Theorem 1 we prove the following.

Theorem 2. *For $\rho = 7/4$ there exist a polynomial time algorithm that given an instance of TAP computes a solution I of size at most the right-hand size of (6). Thus $|I| \leq \frac{7}{4}\tau$.*

3.2 The algorithm (Proof of Theorem 2)

The algorithm as in Theorem 2 follows the line of the algorithms in [8,17], with the major difference that the lower bound is compared versus a fractional solution x and not versus an integral solution. This makes the analysis much more involved.

Let F_L be a minimum w -weight exact cover of L and let $M = \delta_{F_L}(L, L)$ be the set of leaf-to-leaf links in F_L . Initially, assign tokens to the links in M and to the nodes as follows:

Initial token assignment.

- (i) Every link in $M \setminus W$ gets ρ tokens.
- (ii) Every link in $M \cap W$ gets $\rho + \frac{1}{2}$ tokens.
- (iii) Every leaf unmatched by M gets $\rho - \frac{1}{2}$ tokens.
- (iv) Every $v \in R$ gets $\frac{1}{2}x(\delta(v))$ tokens.
- (v) The root r gets 1 token.

By Lemma 2 we have:

Corollary 1. *The total amount of tokens is at most $\rho\tau + 1$.*

At each iteration, the algorithm iteratively contracts a certain subtree T' of T/I , which means the following: combine all nodes in T' into a single node v , delete edges and links with both endpoints in T' , and the edges and links with one endpoint in T' now have v as their new endpoint. For a set of links $I \subseteq E$, let T/I denote the tree obtained by contracting every 2-edge-connected component of $T \cup I$ into a single node. Since all contractions are induced by subsets of links, we refer to the contraction of every 2-edge-connected component of $T \cup I$ into a single node simply as the contraction of the links in I .

We refer to the nodes created by contractions as *compound nodes*. Each compound node always owns 1 token. Non-compound nodes are referred to as *original nodes* (of T). Each time a contraction takes place, the new compound

node gets 1 token, which together with the links added is paid by the tokens owned by the contracted subtree. For technical reasons, r is also considered as a compound node. The non-contracted links in M and original nodes keep their tokens. This means that the algorithm maintains the following invariant:

Tokens Invariant.

- (i) Every link in $M \setminus W$ owns ρ tokens.
- (ii) Every link in $M \cap W$ owns $\rho + \frac{1}{2}$ tokens.
- (iii) Every leaf unmatched by M owns $\rho - \frac{1}{2}$ tokens.
- (iv) Every $v \in R$ owns $\frac{1}{2}x(\delta(v))$ tokens.
- (v) Every compound node owns 1 token.

The algorithm starts with a partial solution $I = \emptyset$ and with $\text{credit}(T/I) = \text{credit}(T)$ being the right-hand side of (6) plus 1. It iteratively finds a subtree T' of T/I and a cover I' of T' , and *contracts T' with I'* , which means the following: add I' to I , contract T' , and assign 1 token to the new compound node. To use the notation T/I properly, we will assume that I is an exact cover of T' , namely, that the set of edges of T/I that is covered by I equals the set of edges of T' (this is possible due to shadow completion).

Definition 4. A contraction of T' with I' is legal if $\text{tokens}(T') \geq |I'| + 1$.

This means that the set I' of links added to I and the 1 token assigned to the new compound node are paid by the total amount of tokens in T' . The tokens owned by T' are not reused in any other way. We do only legal contractions, which implies that at any step of the algorithm

$$|I| + \text{tokens}(T/I) \leq \text{credit}(T) .$$

Thus at the last iteration, when T/I becomes a single compound node, $|I|$ is at most the right-hand side of (6).

Another property of our contracted tree T' is given in the following definition.

Definition 5 (M -compatible tree). Let M be a matching on the leaves of T/I . A subtree T' of T/I is M -compatible if for any $bb' \in M$ either both b, b' belong to T' , or none of b, b' belongs to T' . We say that a contraction of T' is M -compatible if T' is M -compatible.

A simple example of a legal M -compatible contraction is as follows.

Definition 6. Let $uv \in E$ such that both u, v are unmatched by M . Then adding uv to the partial solution I and assigning 1 token to the obtained compound node is called a greedy contraction.

Note that a greedy contraction is always M -compatible, and that if $\rho \geq 1.5$ then it is legal. One of the steps of the algorithm is to apply greedy contractions exhaustively; clearly, this can be done in polynomial time.

We now describe a more complicated type of legal M -compatible contractions. The *up-link* $up(a)$ of a node a is the link au such that u is as close as possible to the root; such u is called the *up-node* of a . Assuming shadow completion, such u is unique and is an ancestor of a . For a rooted subtree T' of T/I and a node $a \in T'$ we say that T' is *a-closed* if the up-node of a belongs to T' (namely, if no link incident to a has its other endnode outside T'), and T' is *a-open* otherwise. For a node set U we let $up(U) = \{up(u) : u \in U\}$. A rooted subtree T' of T/I is *U-closed* if there is no link in E from $U \cap T'$ to $T \setminus T'$. T' is *leaf-closed* if it is $L(T)$ -closed. A leaf-closed T' is *minimally leaf-closed* if any proper rooted subtree of T' is not leaf-closed.

Definition 7 (semi-closed tree). *Let M be a matching on the leaves of T/I . A rooted subtree T' of T/I is semi-closed (w.r.t. M) if it is M -compatible and closed w.r.t. its unmatched leaves. T' is minimally semi-closed if T' is semi-closed but any proper subtree of T' is not semi-closed.*

For a semi-closed tree T' let us use the following notation:

- $M' = M(T')$ is the set of links in M with both endnodes in T' .
- $U' = U(T')$ is the set of unmatched leaves of T' .

Lemma 3 ([8,17]). *If T' is minimally semi-closed then $M' \cup up(U')$ is an exact cover of T' .*

Thus a minimally semi-closed tree T' admits a cover of size $|M'| + |U'|$. This motivates the following definition, that concerns *arbitrary* semi-closed trees, that may not be minimal.

Definition 8 (deficient tree). *A semi-closed tree T' is deficient if $credit(T') < |M'| + |U'| + 1$.*

Summarizing, our algorithm maintains the following additional invariant:

Partial Solution Invariant.

The partial solution I is obtained by sequentially applying a greedy contraction or a legal semi-closed tree contraction with an exact cover.

In the next section we will prove the following key statement.

Lemma 4. *Suppose that $\rho = 7/4$ and that the Tokens Invariant and the Partial Solution Invariant hold for T , M , and I , and that T/I has no greedy contraction. Then there exists a polynomial time algorithm that finds a non-deficient semi-closed tree T' of T/I and an exact cover $I \subseteq E$ of T' of size $|I| = |M'| + |U'|$. Furthermore, such T' can be found even without knowing the LP solution x .*

Algorithm 1: LP-TREE-COVER($T = (V, \mathcal{E}), E$) (A 1.75-approximation algorithm)

```

1 initialize:  $I \leftarrow \emptyset$ 
2  $F_L \leftarrow$  minimum  $w$ -weight exact edge cover of  $L$ ,  $M \leftarrow \delta_{F_L}(L, L)$ .
3 while  $T/I$  has more than one node do
4   Exhaust greedy contractions and update  $I$  and  $M$  accordingly.
5   Find a subtree  $T'$  of  $T/I$  and an exact cover  $I'$  of  $T'$  as in Lemma 4.
6   Contract  $T'$  with  $I'$ .
7 return  $I$ 

```

Algorithm LP-TREE-COVER (Algorithm 1) initiates $I \leftarrow \emptyset$ as a partial cover. It computes a minimum w -weight exact edge cover F_L of L , sets $M = \delta(F_L, F_L)$ and initiates the described credit scheme. In the main loop, the algorithm iteratively exhausts greedy contractions, then computes T', I' as in Lemma 4, and contracts T' with I' . The stopping condition is when I covers T , namely, when T/I is a single node.

It is easy to see that all the steps in the algorithm can be implemented in polynomial time. The credit scheme used implies that the algorithm computes a solution I of size at most ρ times the right-hand size of (6). Hence it only remains to prove Lemma 4.

4 Proof of Lemma 4

In what follows, for a subtree T' of T/I let us use the following notation:

- $M' = M(T')$ is the set of links in M with both endnodes in T' .
- U' is the set of leaves of T' unmatched by M
- U'_0 is the set of the original leaves in U' .
- $L' = L(T')$ is the set of leaves of T' and $S' = S(T')$ is the set of stems of T' .
- $R' = V(T') \setminus (L' \cup S')$ and $\Sigma = \sum_{v \in R'} x(\delta(v))$.
- C' is the set of non-leaf compound nodes of T' (this includes r , if $r \in T'$).

Definition 9 (dangerous tree). A semi-closed tree T' is called dangerous (see Fig. 1) if $|C'| = |S'| = |U'_0| = 0$, $|M'| = 1$, $|L'| = 3$, and if a is the unmatched leaf of T' then there exists an ordering b, b' of the matched leaves of T' such that $ab' \in E$, the contraction of ab' does not create a new leaf, and T' is b -open. If such an ordering b, b' is not unique (namely, if also $ab \in E$, the contraction of ab does not create a new leaf, and T' is b' -open, see Fig. 1(b)), then we will assume that the up-node of b is an ancestor of the up-node of b' .

In the next section we will prove the following key statement.

Lemma 5. Under the assumptions of Lemma 4, any deficient tree is dangerous.

Note that the property of a tree being dangerous depends only on the structure of the tree and existence/absence of certain links in E and M , and thus can

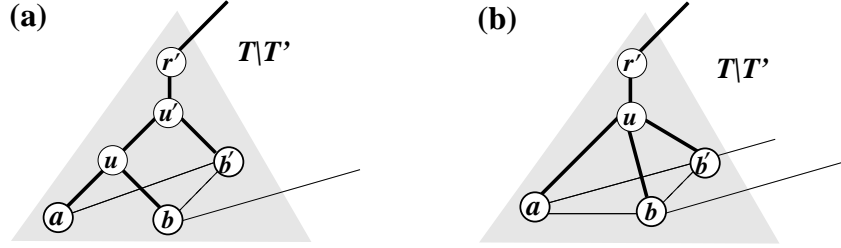


Fig. 1. Dangerous trees. The dashed arc shows the matched pair bb' . Solid thin lines show links that must exist in E . Here u is not a stem and a is a compound node. Some of the edges of T can be paths and $u = r'$ may hold.

be tested in polynomial time. If T/I has a minimally semi-closed subtree T' that is not dangerous, then T' is not deficient, so T' and $I' = M' \cup up(U')$ satisfy the requirement of Lemma 4. Otherwise, if all minimally semi-closed subtrees of T/I are dangerous, then we execute Algorithm 2 below. In the algorithm we define a new matching \tilde{M} obtained from M by replacing the link bb' by the link ab' in each dangerous tree. Note that the property of a tree being semi-closed or dangerous depends on the matching. In what follows, “dangerous” always means w.r.t. the matching M ; for “semi-closed” the default matching is M , and we will specify each time when a tree is semi-closed w.r.t. the new matching \tilde{M} .

Algorithm 2: FIND-TREE($T = (V, \mathcal{E}), E, M$) (Finds a non-dangerous semi-closed tree T' and its exact cover I' of size $|I'| = |M'| + |U'|$, when all minimally semi-closed trees are dangerous.)

- 1 Let \tilde{M} be a matching on the leaves of T/I obtained from M by replacing the link $e = bb'$ by the link $\tilde{e} = ab'$ in every dangerous tree T_0 of T .
 - 2 Let T' be a minimally semi-closed tree w.r.t. the matching \tilde{M} .
 - 3 **return** T' and $I' = \tilde{M}(T') \cup up(\tilde{U}')$, where \tilde{U}' is the set of unmatched leaves of T' w.r.t. \tilde{M} .
-

Lemma 6. *Algorithm 2 finds a non-dangerous (non-minimal) semi-closed tree T' and its cover I' of size $|I'| = |M'| + |U'|$.*

Proof. The statement was essentially proved in [8,17], as our dangerous trees coincide with a similar definition in [8,17]. The key point is that for any dangerous tree T_0 , either all of a, b, b' belong to T' , or none of them belongs to T' . This implies that T' is semi-closed and properly contains some dangerous tree. Consequently, it can be shown that T' is not dangerous. Furthermore, the bijective correspondence between links in M and \tilde{M} implies $|M(T')| = |\tilde{M}(T')|$, and the statement follows. \square

The proof of Lemma 4 is now complete. It remains only to prove Lemma 5.

5 Proof of Lemma 5

Let T' be a deficient tree with root r' , so $\text{tokens}(T') - (|M'| + |U'|) < 1$. We will show that T' is dangerous (under the assumptions of Lemma 4).

In what follows, note that M is a matching on the leaves of T , and by the Partial Solution Invariant, M remains a matching on the leaves of T/I and every leaf of T/I matched by M is an original leaf of T ; this is so since we contract only M -compatible trees. In particular, $x(\delta(b)) = 1$ for every $b \in L \setminus U$, by (3) (note that $L \setminus U$ is the set of leaves of T/I matched by M). Furthermore, the Partial Solution Invariant implies that every stem s in T/I has exactly two leaf descendant, and they are both original leaves.

The amount of tokens owned by T' is:

$$\text{tokens}(T') = \rho|M'| + \frac{1}{2}|M' \cap W| + |U'| + (\rho - \frac{1}{2})|U'_0| + |C'| + \Sigma$$

Thus

$$\text{tokens}(T') - (|M'| + |U'|) = (\rho - 1)|M'| + \frac{1}{2}|M' \cap W| + (\rho - \frac{3}{2})|U'_0| + |C'| + \Sigma.$$

Since T' is deficient, $\text{tokens}(T') - (|M'| + |U'|) < 1$. Thus for $\rho = 7/4$ we have:

$$\text{tokens}(T') - (|M'| + |U'|) = \frac{7}{4}|M'| + \frac{1}{2}|M' \cap W| + \frac{1}{4}|U'_0| + |C'| + \Sigma < 1. \quad (7)$$

Lemma 7. $|C'| = 0$, $|M' \cap W| = 0$, $|M'| \leq 1$, and $|S'| = 0$.

Proof. From (7) we immediately get that $|C'| = 0$, $|M' \cap W| = 0$, $|M'| \leq 1$, and if $|M'| = 1$ then $|U'_0| = 0$. It remains to prove that $|S'| = 0$. Suppose to the contrary that T' has a stem s . Let a, b be the two leaf descendants of s , so a, b are original leaves and ab is a twin link. Since $ab \in W$, $ab \notin M'$. From the assumption that T/I has no link greedy contraction we get that one of a, b is matched by M , as otherwise ab gives a greedy contraction. Moreover, $|M' \cap W| = 0$ and $|M'| \leq 1$ implies that exactly one of a, b is matched by M . This implies the contradiction $|M'| = 1$ and $|U'_0| \geq 1$. \square

Lemma 8. $\Sigma \geq |U'| + 1 - 2|M'|$.

Proof. Note that no link has both endnodes in U' (since T/I has no greedy contraction), and that $\delta(U') \cap \delta(T') = \emptyset$ (since T' is U' -closed). Thus

$$x(\delta(U') \cup \delta(T')) = \sum_{u \in U'} x(\delta(u)) + x(\delta(T')) \geq |U'| + 1.$$

If $e \in \delta(U')$, then e contributes x_e to Σ , unless e is incident to a matched leaf. However, $x(\delta(b)) = 1$ for every matched leaf b , and the number of matched leaves in T' is exactly $2|M'|$. Hence $\Sigma \geq x(E') \geq |U'| + 1 - 2|M'|$, as claimed. \square

Lemma 9. $|M'| = 1$, $\Sigma < 1/2$, and $|L'| = 3$.

Proof. We prove all statements by contradiction. If $|M'| \neq 1$, then $|M'| = 0$, by Lemma 7. This implies $\Sigma \geq |U'| + 1 \geq 2$, by Lemma 8, and we obtain the contradiction $\text{tokens}(T') - (|M'| + |U'|) \geq \frac{1}{2}\Sigma \geq 1$. Thus $|M'| = 1$.

If $\Sigma \geq \frac{1}{2}$, then by (7) we get $\text{tokens}(T') - (|M'| + |U'|) \geq \frac{7}{4}|M'| + \frac{1}{2}\Sigma \geq 1$, contradicting the assumption that T' is deficient.

We show that $|L'| = 3$. Note that $|L'| \geq 4$ is not possible, since then $|U'| \geq 2$, which implies the contradiction $\Sigma \geq |U'| + 1 - 2|M'| \geq 1$. Also, $|L'| = 1$ is not possible, since then $|M'| = 0$. We are therefore left with the case $|L'| = 2$, say $L' = \{b, b'\}$. Then, since $|M'| = 1$, we have $M' = \{bb'\}$. Consequently, the contraction of bb' creates a new leaf. We obtain a contradiction by showing that then the path between b and b' in T/I has an internal compound node. By the Partial Solution Invariant b, b' are original leaves. Note that in the original tree T the contraction of bb' does not create a new leaf, since $bb' \notin W$. This implies that in T , there is a subtree \hat{T} of T hanging out of a node z on the path between b and b' in T . This subtree \hat{T} is not present in T/I , hence it was contracted into a compound node during the construction of our partial solution I . Thus T/I contains a compound node \hat{z} that contains \hat{T} , and since \hat{z} contains a node z that belongs to the path between b and b' in T , the compound node of T/I that contains z belongs to the path between b and b' in T/I , and it is distinct from b, b' , since both b, b' are original leaves. \square

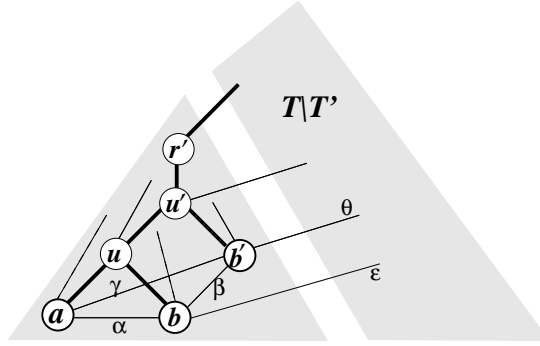


Fig. 2. Illustration to the proof of Lemma 5 (links in E_1 are shown by dashed lines).

Now we finish the proof of Lemma 5. Let bb' be the matched pair and a the unmatched leaf of T' . Let u and u' be the least common ancestor of ab and ab' , respectively, and assume w.l.o.g. that u is a descendant of u' (see Fig. 2). Let $x_{ab} = \alpha$, $x_{bb'} = \beta$, $x_{ab'} = \gamma$, $x(\delta(b, T \setminus T')) = \epsilon$, and $x(\delta(b', T \setminus T')) = \theta$. To finish the proof of the lemma, it is sufficient to show the following:

- If $u \neq u'$, then $\gamma > 0$ and $\epsilon > 0$.
- If $u = u'$, then at least one of the following holds: $\gamma, \epsilon > 0$ or $\alpha, \theta > 0$.

Indeed, if $u \neq u'$, then $\gamma > 0$ implies that the link ab' exists, and $\epsilon > 0$ implies that T' is b -open. Thus, by the definition, T' is dangerous. The same holds if $u = u'$ and $\gamma, \epsilon > 0$. If $u = u'$ and $\alpha, \theta > 0$, then ab exists (since $\alpha > 0$) and T' is b' -open (since $\theta > 0$); thus by exchanging the roles of b, b' we get that T' is dangerous, by the definition. Consider the following links sets:

- $E_1 = \delta(\{a, b\}, R') = x(\delta(a, R')) + x(\delta(b, R'))$ are the links from a, b to R' .
- $E_2 = \delta(T_u \setminus \{a, b\}, T \setminus T_u) = \delta(T_u \cap R', T \setminus T_u)$ are the links from $T_u \cap R'$ to nodes outside T_u .
- $E_3 = \delta(R', T \setminus T')$ are the links from R' to nodes outside T' .

Any $e \in E_1 \cup E_2 \cup E_3$ contributes x_e to Σ . Recalling that $x(\delta(b)) = x(\delta(b')) = 1$ and $x(\delta(a)) \geq 1$, it is easy to verify that:

- (i) $\Sigma \geq x(E_1) + x(E_2) = x(\delta(T_u)) - \gamma \geq 1 - \gamma$ if $u \neq u'$.
The first inequality is since every $e \in E_1 \cup E_2$ contributes x_e to Σ and since $E_1 \cap E_2 = \emptyset$.
- (ii) $\Sigma \geq x(E_1) = x(\delta(a, R')) + x(\delta(b, R')) \geq x(\delta(a, R')) \geq x(\delta(a)) - \alpha - \gamma \geq 1 - \alpha - \gamma$.
- (iii) $\Sigma \geq x(E_3) \geq x(\delta(T', T \setminus T')) - \theta - \epsilon \geq 1 - \theta - \epsilon$.
- (iv) $\alpha + \epsilon \leq 1$ (since $x(\delta(b)) = 1$) and $\gamma + \theta \leq 1$ (since $x(\delta(b')) = 1$).

Suppose that $u \neq u'$. Then $1/2 > \Sigma \geq 1 - \gamma$, by (i); hence $\gamma > 1/2$. If $\epsilon = 0$ then (iii) implies $\theta > 1/2$, and we obtain a contradiction to (iv) $\gamma + \theta > 1$. Thus, $\gamma, \epsilon > 0$ in this case, as claimed.

Suppose that $u = u'$. By (ii) and (iii), $\alpha + \gamma > 1/2$ and $\theta + \epsilon > 1/2$. One can easily verify that combined with (iv) this gives that we must have $\gamma > 0$ and $\epsilon > 0$, or $\alpha > 0$ and $\theta > 0$, as claimed.

This concludes the proof of Lemma 5.

References

1. J. Cheriyan, Z. Gao, A. Linhares, and C. Swamy. Private communication. 2014.
2. J. Cheriyan, T. Jordán, and R. Ravi. On 2-coverings and 2-packing of laminar families. In *ESA*, pages 510–520, 1999.
3. J. Cheriyan, H. Karloff, R. Khandekar, and J. Koenemann. On the integrality ratio for tree augmentation. *Operation Research Letters*, 36(4):399–401, 2008.
4. Y. Chu and T. Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400, 1965.
5. N. Cohen and Z. Nutov. A $(1 + \ln 2)$ -approximation algorithm for minimum-cost 2-edge-connectivity augmentation of trees with constant radius. *Theoretical Computer Science*, 489-490:67–74, 2013.
6. J. Edmonds. Optimum branchings. *J. Res. Nat. Bur. Standards*, pages 233–240, 1967.
7. G. Even, J. Feldman, G. Kortsarz, and Z. Nutov. A $3/2$ -approximation for augmenting a connected graph into a two-connected graph. In *APPROX*, pages 90–101, 2001.

8. G. Even, J. Feldman, G. Kortsarz, and Z. Nutov. A 1.8-approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2. *ACM Transactions on Algorithms*, 5(2), 2009.
9. G. N. Frederickson and J. Jájá. Approximation algorithms for several graph augmentation problems. *SIAM J. Computing*, 10:270–283, 1981.
10. G. N. Frederickson and J. Jájá. On the relationship between the biconnectivity augmentation and traveling salesman problem. *Theoretical Computer Science*, 19(2):189–201, 1982.
11. M. Goemans, A. Goldberg, S. Plotkin, E. T. D. Shmoys, and D. Williamson. Improved approximation algorithms for network design problems. In *SODA*, pages 223–232, 1994.
12. M. Goemans and D. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Computing*, 24(2):296–317, 1995.
13. K. Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.
14. S. Khuller. Approximation algorithms for finding highly connected subgraphs (chapter 6). In *Approximation algorithms for NP-hard problems* (Ed. D. S. Hochbaum). PWS, Boston, 1996.
15. S. Khuller and R. Thurimella. Approximation algorithms for graph augmentation. *J. of Algorithms*, 14:214–225, 1993.
16. G. Kortsarz and Z. Nutov. Approximating minimum cost connectivity problems (chapter 58). In *Handbook of Approximation Algorithms and Metaheuristics* (Ed. T. F. Gonzales). Chapman & Hall/CRC, 2007.
17. G. Kortsarz and Z. Nutov. A simplified 1.5-approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2. Manuscript, 2014.
18. L. C. Lau, R. Ravi, and M. Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, 2011.
19. Y. Maduel and Z. Nutov. Covering a laminar family by leaf to leaf links. *Discrete Applied Mathematics*, 158(13):1424–1432, 2010.
20. H. Nagamochi. An approximation for finding a smallest 2-edge connected subgraph containing a specified spanning tree. *Discrete Applied Mathematics*, 126:83–113, 2003.
21. A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency*. Springer-Verlag Berlin, Heidelberg New York, 2004.
22. A. Sebo and J. Vygen. Shorter tours by nicer ears: $7/5$ -approximation for the graph-TSP, $3/2$ for the path version, and $4/3$ for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, 2014.